Windows & .NET Magazine Network

Home | Hot Topics | Recent Articles | Publications Archive | Forums | Events | My Account

Keyword Search GO InstantDoc ID GO Advanced Search LOG ON

Driving the Enterprise Network ▪ NT ▪ 2000 ▪ XP ▪ 2003

Related Hot Topics: **Active Directory | Active Directory | Exchange & Outlook**

Mail Attender®
TOTAL EMAIL MANAGEMENT
CLICK HERE FOR A FREE TRIAL!
• IMPLEMENT CORPORATE POLI
• ENFORCE RETENTION RULES
• SEARCH & RETRIEVE CONTENT
• COMPLETE MANAGEMENT FOR FILES, MAILBOXES & PUBLIC F

[Feature]

How to Customize DS-to-AD Attribute Synchronization

Kieran McCorry

InstantDoc #19712
March 2001

ADC attribute mapping

An important part of Active Directory Connector (ADC) functionality is the bidirectional synchronization of objects between the Microsoft Exchange Server 5.5 Directory Service (DS) and Active Directory (AD). When the ADC synchronizes an object from the DS to AD, it synchronizes attributes associated with the Exchange Server 5.5 object (e.g., address, telephone number) to their attribute counterparts in the AD object.

Although you can control to some extent which attributes you synchronize from one directory to another, you can do little with the ADC's out-of-the-box UI functionality to control the mapping of individual attributes. However, you can use some techniques to customize attribute mapping, especially from the Exchange Server 5.5 DS to AD. Let's look at how you can refine ADC behavior for your environment. The Exchange 2000 Server documentation doesn't include information about these techniques, although the Microsoft articles I cite in this article describe some customizing techniques. However, the techniques are in line with the standard operation of the product.

Default Attribute-Mapping Behavior

The Microsoft Management Console (MMC) Active Directory Connector Manager snap-in offers Microsoft's predefined list of attributes that you can either include in the synchronization activity or not. By default, most Exchange Server 5.5 mailbox or custom recipient attributes that you want replicated to AD are synchronized automatically when you establish a connection agreement (CA) between the source container and the target organizational unit (OU). Useful attributes include naming and addressing information, telephone and location details for the Global Address List (GAL), and mailbox properties such as the Home MDB (which indicates the server on which a user's mailbox resides) that are necessary for Exchange to deliver mail. These default attributes are sufficient for most installations.

However, in some cases, you need to modify the defaults to address specific circumstances in your situation. To control which attributes you want synchronized, you need to modify the Default ADC Policy. To modify the policy, right-click Active Directory Connector Management in the Active Directory Connector Manager snap-in, select Properties, then click either the From Windows or the From Exchange tab, depending on the flow direction you want. Figure 1, page 2, shows the From Exchange interface. Select the check boxes for the attributes you want to synchronize. Typically, you might want only to suppress synchronization of an attribute, such as a custom attribute that holds some legacy information whose value you don't want to synchronize to AD because the value of the AD attribute is coming from some other directory source.

Moving Beyond the Schema-Mapping GUI

As you can see from Figure 1, the Active Directory Connector Manager gives you few options. If you want to perform more complex attribute synchronizations, such as mapping the value of one attribute in the source directory to a different attribute in the target directory, you need a different method. You need to bypass the snap-in and manipulate the policy directly by using ADSI Edit to change settings in AD.

You can install ADSI Edit from the Support directory on the Windows 2000 CD-ROM. Figure 2 shows how the Default ADC Policy appears as an object under the Active Directory Connections object in AD. For more information about ADSI Edit, see Tony Redmond, "Introducing the ADSI Edit Utility," July 2000.

Because you define attribute-mapping rules on the Default ADC Policy, the ADC enforces these rules for all CAs that are homed on a particular ADC. (For more information about CAs, see Bill English, "Planning for and Configuring the Active Directory Connector," March 2000, and Tony Redmond's *Windows 2000 Magazine* article "The Active Directory Connector," January 2000.) However, you can be more specific with attribute-mapping rules by setting individual mapping rules for each CA. Attribute - mapping rules from the Default ADC Policy and from individual CAs merge during a synchronization run.

Modifying the Mapping Tables

You can define the attribute mapping rules on two attributes of the Default ADC Policy or a CA. The *msExchServer1SchemaMap* and *msExchServer2SchemaMap* attributes define attribute-mapping rules for AD to the Exchange Server 5.5 DS and the Exchange Server 5.5 DS to AD, respectively. You can find these attributes associated with defined CAs under the Active Directory Connections object in AD. Figure 3 shows an example of what you see when you look at the properties of the policy object and specifically at the msExchServer1SchemaMap. The string in the Value(s) text box in Figure 3 is just the first few characters of a string that continues for some 18,029 characters, or 269 individual lines in the attribute-mapping rule table. The ADC sets these rules upon installation. The rules reflect the settings that Microsoft believes are optimal for synchronizing attribute data to and from the Exchange Server 5.5 DS and AD. As you can see, large text strings that control application behavior reside in AD. In earlier software versions, such settings resided in .ini files: AD provides a much cleaner solution.

To customize the mapping rules, first click Clear to display the mapping rule in the Edit Attribute text box; you can't edit the rule when it's in the Value(s) text box. After you've changed the rule to reflect your requirements, click Set to write the new rule to AD and activate it. I find it challenging to edit these rules by changing the text string in the Edit Attribute text box. An easier method is to copy the text string from the Value(s) text box and paste it into your favorite editor (e.g., Notepad, Wordpad), in which the embedded carriage return/line feeds (CRLFs) make the text more readable. Using an editor, you can more easily find the rule you want to modify and understand other related rules. However, you can't use an editor to make changes to the mapping rules, then paste the new rules back into AD. You must make changes directly to AD by editing the rule in the long text string.

When you install the ADC, the attribute-mapping tables are populated from predefined text files on the Exchange 2000 Server distribution CD-ROM. Two files—local.map and remote .map—are in the \ADC\i386 directory. The local.map file contains the complete set of mapping rules that populate the msExchServer2SchemaMap attribute, and the remote.map file defines the rules for the msExchServer1SchemaMap. If you know which modifications you need to make to the attribute-mapping files, you can edit these text files (providing you copy the installation kit onto writable media) before you install the ADC. Editing the files before installation is particularly useful, and efficient, if you're deploying multiple ADCs and you need to make consistent modifications to the mapping tables. Because the mapping table resides in AD's Configuration Naming Context, it's replicated across the whole forest. If you've already installed the ADC and you install it in another location with some modifications to the mapping tables, the Default ADC Policy won't include the modifications. However, the mapping tables specific to new CAs that you create will use the new settings.

In the unlikely event that you need to restore the mapping tables to their default configurations, you must reinstall the ADC. Any changes that you make to the attribute mappings affect only the mapping tables in AD, and an ADC reinstall normalizes AD tables by reading the original set of mapping rules from local.map and remote.map. Using ADSI Edit to make changes to the mapping tables can introduce errors. No validation takes place as you modify the text string, and any errors can cause attribute synchronization to stop for that CA. If you're serious about providing attribute mappings and think you'll need to make significant changes regularly, consider writing an Active Directory Service Interfaces (ADSI) or Lightweight Directory Access Protocol (LDAP) utility.

Changing Attribute Mappings

Editing the schema-mapping file lets you change how the value of one attribute is mapped to another. For example, you might want to change the mapping between the City and the Office attributes as you synchronize Exchange Server 5.5 DS objects to AD. Figure 4 shows Sharon Stafford's Exchange Server 5.5 mailbox. When the ADC synchronizes this mailbox into AD, it maps the Exchange Server 5.5 Office attribute to the Office attribute of the object in AD. However, in this case, let's say I'd rather have the value of the Exchange Server 5.5 City attribute written to the AD Office attribute. The default mapping rules specified in msExchServer2SchemaMap define the value of the Exchange Server 5.5 Office attribute to map directly to the value of the AD Office attribute. The sidebar "Attribute-Mapping Rule Syntax," page 4, explains the fields in the rules, and the sidebar "Exchange Server 5.5 and LDAP Names," page 5, explains the object names that appear in the rule. The appropriate rule (let's call it Rule 1) is

```
local###physicalDeliveryOffice_
Name#physicalDeliveryOffice_
Name###0#
```

This rule states that for all objects in the Exchange Server 5.5 DS that the ADC processes, map the Exchange Server 5.5 Office attribute to the AD Office attribute. Similarly, another mapping rule in the mapping table maps the Exchange Server 5.5 City attribute value to the AD City attribute. The rule (let's call it Rule 2) used to effect this mapping is

```
local###l#l###0#
```

The City attribute is difficult to see in this rule. As the sidebar "Exchange Server 5.5 and LDAP Names" explains, the lowercase letter *l*—short for location—represents the City attribute in LDAP terminology; there's no specific attribute called *City*. To make the AD Office attribute take on the value of the Exchange Server 5.5 City attribute, I change Rule 1 to read

```
local###l#physicalDeliveryOffice
Name###0#
```

You don't need to change Rule 2 because the ADC can map one Exchange Server 5.5 attribute to multiple attributes in AD. Thus, when the ADC synchronizes the Exchange Server 5.5 mailbox that Figure 4 shows, the mailbox's Office value in AD will be Dublin, not Belfield, as Figure 5 shows. Furthermore, the mailbox's AD City attribute will have a value of Dublin, so this synchronization has performed a one-to-many mapping of attribute data.

Dealing with Conflicting Mapping Rules

The mapping schema doesn't use mapping-rule precedence; that is, if you have two rules that map to the same target attribute, the latter rule won't override the former, nor will the former override the latter. In this example, if you have two rules trying to write to the *physicalDeliveryOfficeName* target AD attribute, the entire synchronization operation for this object will fail. Figure 6, page 6, shows the event that appears in the event log when you have two mapping rules directed to the same target attribute.

In this case, the ADC returned the Constraint Violation error because physicalDeliveryOfficeName is a single-valued attribute and the ADC processes both mapping rules and attempts to assign both values to the target attribute. This process results in an error and cancellation of the synchronization operation for this object. However, for multivalued attributes, multiple mapping rules can target the same attribute successfully.

When Mapping Table Changes Take Effect

When you make a change to the attribute-mapping table, changes in mapping policy take effect the next time the ADC initiates synchronization. If the ADC creates a new object during a synchronization run, it will honor the new mappings but won't apply the new rules to update objects that have already been synchronized into AD.

The ADC applies updates in line with the new attribute-mapping rules to already-synchronized objects only if some change on the source object causes the ADC to process it again for the synchronization. Such a change would include updating a telephone number attribute or an address field, for example, but it wouldn't include an action such as mapping the Office attribute to the City attribute. As long as you make some change to the source attribute, you can apply changes by using a comma separated values (CSV) file and the Exchange Server 5.5 admin /I option. The same rules apply to attribute-mapping operations in the reverse direction (i.e., from AD to the DS).

Although I've manipulated the Office and City fields in this example, you can apply these rules to all attributes associated with the directory objects. In most cases, you'll map the attribute values as is, but other attributes (e.g., custom attributes) deserve special attention because organizations have used them in different ways for different reasons. And with the introduction of AD and the way in which organizations are using it, you'll probably want to redefine some custom attribute mappings.

AD Distinguished Name Mapping

Objects that the ADC creates in AD have a distinguished name (DN) that is built from a combination of the AD container into which the object is being created and an Exchange Server 5.5 attribute of the source object. For example, for Sharon's mailbox in Figure 5, the default DN that is built in AD (let's call it *DN 1*) is

```
CN=Sharon Stafford,CN=Users,
DC=research,DC=compaq,DC=com
```

The least-significant relative distinguished name (RDN) part of the DN (i.e., CN=Sharon Stafford) comes directly from the Exchange Server 5.5 mailbox Display Name attribute. If the OU has more than one Sharon Stafford, the ADC eliminates the problem of duplicate names by simply appending a numerical value to the Display Name. In the Exchange Server 5.5 DS, the LDAP name for the Display Name is *cn*. Accordingly, a special rule in the msExchServer2SchemaMap table explicitly defines the RDN for any AD objects that the ADC will create. This rule is the last rule on the mapping table and is defined as follows:

```
local###cn#Override_RDN_
Value###140#
```

The syntax of this rule is slightly different from the other rules in the mapping table. No AD attribute appears in the rule, but the Override_RDN_Value string is a directive that tells the ADC to modify the RDN of the ADC-created object. In this rule, *140* represents the mapping flag. The sidebar "Attribute-Mapping Rule Syntax" explains the function of mapping flags.

You can modify this rule to change the source attribute for the RDN. For example, you can replace the source *cn* attribute with the *sn* attribute, or possibly with the *mailNickname* attribute. Replacing the source *cn* attribute in this way forces the ADC to build the RDN by using the surname or the Exchange Server 5.5 mailbox alias, respectively. Similarly, some organizations hold badge numbers or employee IDs in a custom attribute. You can use these source attributes to build the DN of the object in AD.

This rule affects existing objects in AD as well as new objects that the ADC creates in AD. For example, let's say that you had migrated a Windows NT 4.0 user account to a Win2K account by using a tool such as the Microsoft Active Directory Migration Tool (ADMT) before you started using the ADC. When ADMT creates the new user object in Win2K, it uses the NT SAM account name (e.g., Stafford) to build the least-significant RDN part of the DN. This action creates an object in AD with the following DN (let's call it *DN 2*):

```
CN=stafford,CN=Users,DC=
research,DC=compaq,DC=com
```

The ADC matches the Exchange Server 5.5 mailbox object with the existing account in AD (i.e., it matches the Exchange Server 5.5 mailbox Assoc-NT-Account attribute with the sIDHistory attribute of the AD object). When the ADC makes the match, in addition to synchronizing attributes from the Exchange Server 5.5 mailbox into the existing AD object, the RDN override rule forces the AD object's DN to be updated from DN 2 to become based on the Exchange Server 5.5 Display Name attribute. Ultimately, the AD object's DN becomes DN 1.

Although this RDN mapping yields aesthetically pleasing DNs, some administrators dislike it. Reasons for their dislike include the confusion associated with DNs that change during the course of the migration, the possibility of duplicate Exchange Server 5.5 Display Names—which results in appending *-1* to the DN—and the potential effect of applications or processes that rely on consistent DNs.

In any event, you might want to either disable the RDN mapping rule by using the 0x10 flag (Table A in the sidebar "Attribute-Mapping Rule Syntax" explains mapping flags) or map it to an attribute consistent with the existing RDN structure—perhaps by using the Exchange Server 5.5 mailbox alias if it corresponds to the NT SAM account name. (You can also disable the RDN mapping by setting the msExchServer1Flags attribute to the value of 2. The Microsoft article "XADM: ADC Overwrites Display Name with Exchange Server 5.5 Display Name" at http://support.microsoft.com/support/kb/articles/q269/8/43.asp discusses potential problems with this operation.)

A Good Technique to Know

In many environments, you don't need to interfere with the default ADC attribute mapping. Attributes in the Exchange Server 5.5 DS have natural counterparts in AD, and mappings between the two environments are predefined when you install the ADC. Some unique circumstances might require a mapping modification here or there, especially if an organization has made heavy use of custom attributes.

Although you might not need to change mapping rules, you might want to change the way in which the ADC creates or modifies the DN for objects in AD. Being able to suppress DN modification or map it by using a different source attribute is sure to be an advantage for anyone responsible for integrating the old Exchange Server 5.5 DS with the new AD.

Figure 1

TABLE A: Flag Descriptions for the Attribute-Mapping Rules

Flag	Description
0x1	Used when the source attribute is multivalued but the target attribute is single-valued; causes the first source value to be mapped into the target attribute.
0x2	Used when the source attribute has a DN syntax but the ADC can't find the DN in the target directory. In such cases, the source value is written to the unmerged attributes list for later use when the DN can be resolved.
0x4	Used when the source attribute is single-valued but the target attribute is multivalued; causes the source value to be written into the first target value.
0x8	Used when the source attribute is multivalued and the target attribute is single-valued; causes all values from the source attribute to be written as a single value in the target attribute or as a comma-separated list.
0x10	Disables the mapping rule.
0x20	Used when the source attribute is a custom attribute used for mapping purposes and isn't exposed in the directory schema. Reserved for internal ADC use.
0x40	Used when the target attribute is a custom attribute used for mapping and isn't exposed in the directory schema. Reserved for internal ADC use.
0x100	Hides the mapping rule from the Microsoft Management Console (MMC) Active Directory Connector Manager snap-in interface.
0x200	If the CA allows, merges the source value into the target attribute rather than overwriting.
0x400	If the source value is of the DN syntax type and the link can't be resolved, adds the value into the Exchange 2000 Server unmerged-attributes list.